Routing for the Pennsylvania Maglev System by a Genetic Algorithm

Yong-Yan Cao[†], Zongli Lin[†], Theo C. Giras[†] and Daniel R. Disk[‡]

[†] Center of Rail Safety-Critical Excellence
Department of Electrical and Computer Engineering
University of Virginia
P.O. Box 400743, Charlottesville, VA 22904-4743
Email: yycao, zl5y, tgiras@virginia.edu

[‡] Director of Engineering, MAGLEV Inc.,
1905 Industry Road, Box 11, McKeesport, PA 15132
Tel: (412)948-1905, Fax: (412) 948-1051
Email: drdisk@maglevinc.com

Abstract
The scheduling of vehicle movement activities is an important component in high-speed railway transit operations planning process. This paper presents a mathematical programming approach to vehicle routing for the Pennsylvania Transrapid maglev train systems. A mathematical model of the routing procedure is first derived. Based on this model, the vehicle routing problem is formulated into a constrained optimization problem, in which the train miles traveled are maximized subject to various operational and safety requirements. This large-scale nonlinear optimization problem is then solved by a genetic algorithm.

1. Introduction

As transportation evolves in the 21st century, a novel and innovative high-speed ground transportation system, the Transrapid Maglev System, is being deployed as the "Pennsylvania Project" for use in both passenger and light freight service in the Pittsburgh, Pennsylvania region. The system consists of a 47-mile dual guideway with 32 propulsion segments, 15 switches and 4 MAGPort^R stations [1]. Multiple vehicles are to be operated simultaneously on the system. Key to the safe and efficient operation of the Transrapid Maglev System is a vehicle management algorithm (VMA). An important component of the VMA is a vehicle routing scheduler that generates an optimal vehicle routing schedule that respect all operational and safety requirements. The optimality is usually with respect to the train miles traveled, an indication of the utilization of the system. Such a vehicle routing scheduler should also be computationally feasible for a rapid generation of new optimal routing schedules following the reconfiguration of the guideway system due to, for example, the failure of a guideway segment.

This paper proposes a mathematical programming approach to the development of a vehicle routing scheduler in conjunction with the development of a proof-of-concept VMA for the Pennsylvania Transrapid Maglev System [1]. Although the development of the vehicle routing scheduler is presented in the context of the Pennsylvania Transrapid Maglev System

[1] After this analysis work was finished, the project was changed due to the environmental processes. Today the project is 54 miles long and there are now two stations in the area of the airport so there are now a total of 5 stations. There also is an additional substation and an increased number of propulsion blocks.

currently under deployment, it is applicable to other Transrapid Systems to be deployed in elsewhere. The development of the vehicle scheduler consists of the development of a mathematical model of the vehicle routing procedure, the formulation of the routing problem into a constrained optimization problem and the solution algorithm of such an optimization problem. The mathematical model will include various aspects of the operation of the system including the computation of the miles traveled by a particular train and the guideway segment out-of-service constraints. The optimization objective function will be the train miles traveled over a given time period. The optimization constraints include the avoidance of deadlock, guaranteed headway, and minimum MAGPort^R dwell times. Due to the large number of hard constraints, the resulting optimization problem is highly nonlinear and of large scale. We will resort to genetic algorithms (GAs) for the solution of this optimization.

Genetic algorithm is a non-gradient-based optimization procedure based on Darwinian's survival of the fittest mechanisms originated by Holland [2]. It turned out to be a powerful tool in the field of global optimization [2,3,4]. It has been applied successfully to real-world problems and exhibited, in many cases, better search efficiency than traditional optimization algorithms. Compared to traditional search and optimization procedures, such as calculus-based and enumerative strategies, genetic algorithms are robust, global and generally more straightforward to apply in situations where there is little or no a priori knowledge about the process to be controlled. As it does not require derivative information or a formal initial estimate of the solution region and because of the stochastic nature of the search mechanism involved, a genetic algorithm is capable of searching the entire solution space with more likelihood of finding the global optimum.

The remainder of the paper is organized as follows. Section 2 develops a mathematical model of the routing procedure and formulates the vehicle routing problem into a constrained optimization problem. Section 3 presents a genetic algorithm that solves the optimization problem formulated in Section 2. Section 4 presents case studies to demonstrate the application and effectiveness of the proposed routing approach. These case studies include both hazard free and hazardous guideway scenarios. Section 5 concludes the paper.

2. Mathematical Model and the Optimization Problem

This section will first develop a mathematical model of the vehicle movement in the Pennsylvania Maglev System. Based on this mathematical model, the vehicle routing problem is formulated into a constrained optimization problem. Both the optimization objective function and the constraints will be derived.

The dual guideway configuration of the Pennsylvania Transrapid Maglev System is shown in Figure 2-1.

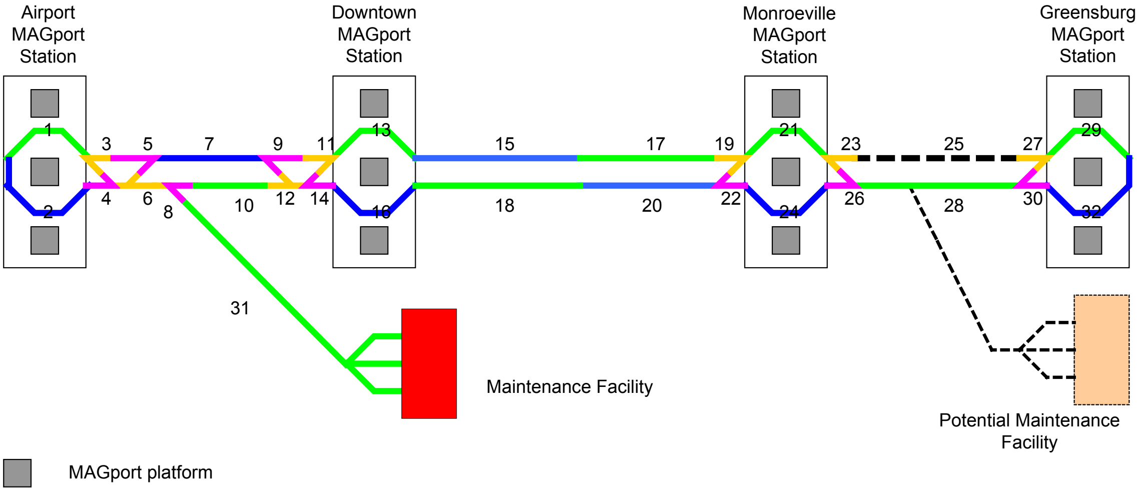

Figure 2-1: Pennsylvania Project Dual Guideway

2.1 Notation

Number of vehicle	N_v	
Number of guideway segment	N_s	
Time zone	T_f	$= T \times T_0$
Sample time	T_0	= 0.5 minute
Set of propulsion segment Set of station segment Set of switch segment Set of connection segment	S_{pr} S_{st} S_{sw} S_{co}	$= [1, \ldots, N_s]$ $S_{pr} = S_{st} \cup S_{sw} \cup S_{co}$
Maximum speed of segment *i*	$V_m(i)$	(= 400km/h =400/60km/m)
Speed of vehicle *i* at time *t*	$s(i, t)$	
Length of segment *i*	$l(i)$	If $i \in S_{st} \cup S_{sw}$, then set $l(i) = 0$
Minimum dwell time at station *i*	$T_{dw}(i)$	If *i* is not a station, $T_{dw}(i) = 0$
Dwell time at station *i* at time *t*	$t_{dw}(i,t)$	
Minimum headway time of station *i*	$T_{hw}(i)$	If *i* is not a station, $T_{hw}(i) = 0$
Headway time of station *i* at time *t*	$t_{hw}(i,t)$	
Distance vehicle *i* has run at time *t*	$d(i, t)$	
Distance vehicle *i* has run in current segment at time *t*	$d_{cur}(i,t)$	$d_{cur}(i, t) - d_{cur}(i, t-1) = s(i, t)T_0$
Position of vehicle *i* at time *t*	$p(i,t)$	
Position matrix of vehicle *i* in segment *j* at time *t*	$y(i,j, t)$	

2.2 Computational Formulae

The distance that vehicle *i* has run at time *t*:

$$d(i,t)=\sum_{\tau=0}^{t}s(i,\tau)T_0,$$
$$d(i,t)-d(i,t-1)=s(i,t)T_0.$$

The distance that vehicle *i* has run in the current propulsion segment at time *t*:

$$d_{cur}(i,t)=\begin{cases}d_{cur}(i,t-1)+s(i,t)T_0 & \text{if} \quad p(i,t)=p(i,t-1)\\ 0 & \text{otherwise}\end{cases}$$

The dwell time of vehicle *i* in the current propulsion segment at time *t*:

$$t_{dw}(i,t)=\begin{cases}t_{dw}(i,t-1)+T_0 & \text{if} \quad p(i,t)=p(i,t-1)\\ 0 & \text{otherwise}\end{cases}$$

The headway time that a vehicle has arrived at the station *i* at time *t*:

$$t_{hw}(i,t)=\begin{cases}0 & \text{if} \quad p(j,t)=i,\, p(j,t-1)\neq i,\ \forall j\in[1,\cdots,N_v]\\ t_{hw}(i,t-1)+T_0 & \text{otherwise}\end{cases}$$

The position of vehicle *i* at time *t*:

$$p(i,t)=\begin{cases}p(i,t-1) & \text{if} \quad s(i,t)T_0<l(p(i,t-1))-d_{cur}(i,t-1),\\ & \text{or} \quad t_{dw}(i,t-1)<T_{dw}(p(i,t-1)),\\ & \text{or} \quad p(i,t-1)+1=p(j,t-1),\ \exists j\in[1,\cdots,N_v],\, j\neq i\\ & \text{or} \quad t_{hw}(p(i,t-1)+1,t)<T_{hw}(p(i,t-1)+1),\\ p(i,t-1)+1 & \text{otherwise}\end{cases} \tag{1}$$

where $p(i, t) + 1$ denotes the next segment number of $p(i, t)$. If $p(i,t-1)+1\in S_{sw}$, then $p(i, t) + 1$ is set to the number of the next connected station or propulsion segment. Also, only the third vehicle in segment 20 goes to segment 24, the other two vehicles in segment 20 go to segment 21 of Monroeville Station.

The position matrix is defined by

$$y(i,j,t)=\begin{cases}1 & \text{if} \quad p(i,t)=j\\ 0 & \text{otherwise}\end{cases}$$

The guideway is subject to "Out of Service" constraints. In a hazard-free enviroment, the vehicle movement is restricted to a counterclockwise direction. In general, segment 25 will be in out of service, this means all of the vehicles going to segment 25 will have to switch to segment 28 from segment 29 by switch 27, and then switch to segment 21 by switches 23 and 26. In the hazard-free environment,, i.e. there is no other out of service segment, the next segment of segment 29 should be segment 28 and then segment 21. In this environment, we set that any vehicle occupied segment 28 will also occupy segment 25. We have the following mathematical model

$$\begin{cases}y(i,25,t)=1 & \text{if} \quad p(i,t)=28,\\ y(i,28,t)=1 & \text{if} \quad p(i,t)=25.\end{cases}$$

In other out of service situations, we have similar setting. For example, if segment 29 of Greensburg Station is out of service, then the vehicle from segment 32 will not go to segment 29 but go to segment 28 inversely. In this situation, we need to set that any vehicle occupying segment 32 also occupies segment 29, i.e.,

$$y(i,29,t)=1 \quad \text{if} \quad p(i,t)=32,$$

and the next segment of 32 is changed to segment 28 from segment 29.

Other out of service situations include that any one of the segments 15, 17, 18 and 19 is out of service. For example, if segment 18 is out of service, then any vehicle that occupies segment 15 or 17 will also occupy segments 17, 18 and 20.

2.3 Optimization Objective Function

The objective function to be maximized is the vehicle miles traveled. The optimization problem can then be formulated as

$$\max\left(L=\sum_{i=1}^{N_v}\sum_{t=0}^{T}s(i,t)T_0\right) \tag{2}$$

subject to the following constraints:

1. Constraints on the vehicle speeds

$$\begin{cases} s(i,t)=0 & \text{if} \quad p(i,t)\in S_{st} \\ s(i,t)\le V_m(p(i,t)) & \text{if} \quad p(i,t)\in S_{co} \end{cases}$$

2. Constraint on the current distance

$$d_{cur}(i,t)\le l(p(i,t)),$$

3. Constraint that only one vehicle shall occupy any given propulsion segment at a given time:

$$\sum_{i=1}^{N_v} y(i,j,t)\le 1, \quad \forall j\in S_{ps}.$$

The variables of the above optimization problem are the speed $s(i, t)$, the dwell time in every station $T_{dw}(i)$.

3. Real Coded Genetic Algorithm

Genetic algorithms have proven to be effective tools for solving complex optimization problems in engineering applications. Genetic algorithms are stochastic search techniques based on the mechanism of natural selection and natural genetics. They have also been applied to solve the vehicle routing problems with complex constraints, such as the time deadlines problem, time windows problem, and multiple-depot problem [7,8]. The general procedure of a genetic algorithm can be described as follows.

GA Procedure

1. Generate the initial population of N random solutions at generation $t = 0$;
2. Evaluate the fitness values for each individual in the population;
3. While (not in termination condition) do
   - Select parents from the population,
   - Recombine parents to yield children,
   - Evaluate fitness values of the children, and
   - Select new population for next generation from parents and children.
4. End.

The first step in designing a GA for a particular problem is to devise a suitable representation scheme. This is very important because the rest of the GA depends on the representation. In the early stages of the development, design variables were represented in the binary format [2-5]. Although they have been successfully applied to solve complex optimization problems, binary-coded GAs have some drawbacks in tackling continuous problems [5,6]. In addition, the domains of some design variables may usually be unknown, and therefore we cannot specify the upper and lower bound of the design variables in advance. This makes binary-coded GAs susceptible in tackling the larger or unknown domain, which is needed for determining the length of the chromosome. Therefore, the use of a real coding is considered more appropriate. (We need a couple of sentence describing real-coded GAs).

The solution space of the combinatorial problem with constraints can be divided into a feasible area and an infeasible area. For highly constrained problems, the infeasible area may cover a big portion of the solution space and it is difficult to find feasible solutions. In this case, by dealing with infeasible solutions, the optimization can be performed more rapidly and can produce a better final solution than if the search is limited to only the feasible area of the solution space.

In genetic algorithms, three ways are introduced to deal with the chromosomes that violate the constraints: rejection methods, repairing methods, and penalty methods. The repairing strategy is adopted to deal with the infeasible chromosomes in this paper. The initial population of chromosomes is generated randomly and such randomly generated chromosomes may not satisfy the constraints of the problem. For example, for the vehicle at a station, if the time the vehicle stays in the station is less than the required headway time, the vehicle should still stay at the station, i.e., the speed should be 0 at this time. However, the randomly generated variable is very seldom to be 0. Our method is to set it to be 0 when the time the vehicle stays in the station is less than the required headway time. The chromosomes are also repaired similarly for the other constraints including the speed limit of every propulsion segment. A constraints-checking and -repairing step is inserted in our GA before the objective values are calculated to ensure that all new chromosomes including those generated after application of the genetic operators (crossover, mutation, and merging) satisfy all the constraints.

GAs start searching the solution by initializing population of random candidates to the solution. Every individual in the population undergoes genetic evolution through crossover and/or mutation. Crossover and mutation are two genetic operators used in GAs. For binary-coded GAs the crossover is carried out by swapping the chromosome contents at the crossover site [2,3,4,5]. For real-coded GAs, several types of crossover operator have also been introduced in [4,5,6]. In this paper, the crossover operator is performed as follows. For two parent chromosomes P_1 and P_2, the resulting offspring after crossover are

$$O_1 = P_1 \times \alpha(P_2 - P_1),$$
$$O_2 = P_2 \times \alpha(P_1 - P_2),$$

where is a scaling factor chosen uniformly at random over some interval, typically [0.25, 1.25]. The role of mutation is often seen as providing a guarantee that the probability of searching any given string will never be zero and acting as a safety net to recover good genetic material that may be lost through the action of selection and crossover [3]. Real-coded GAs can take advantage of higher mutation rates than binary-coded GAs, increasing the level of possible exploration of the search space without adversely affecting the convergence characteristics [9,4].

After crossover and mutation, the new population is created based on the previous individuals through a certain selection procedure. The selection procedure is conducted based on the fitness of each individual. In our problem, the stochastic universal sampling selection procedure is applied in conjunction with the elitist strategy to ensure that the best individual in each generation is passed to the next generation. Stochastic universal sampling is a single-phase sampling algorithm with minimum spread and zero bias [4]. After performing selection, crossover and mutation, a number of new individuals are inserted into the population in every generation to replace old members in the population randomly. The number of the new individuals to be inserted is taken to be 10% of the population size. The best individual in the final generation is then chosen as the solution to the problem.

4. Case Study

In this section, we will present the computational results for several possible hazard-free and hazardous cases by solving the optimization problem (2) with the genetic algorithm described in Section 3.

4.1 USE Case I

Our first example is the routing schedule of the hazard-free case. We will show that our genetic algorithm obtains the best result, which supports deadlock and mishap-free system operations when no hazards are present. In this experimental environment, the following system constraints are considered:

Maglev System Operational Constraints

1. Only one vehicle shall occupy any given propulsion segment at a given time.
2. Minimum vehicle headway shall be five (5) minutes.
3. Minimum MAGPort^R Station dwell time shall be 30 seconds.
4. All propulsion segments, including switches and MAGPort^R Stations, are in service.
5. All traffic flow is counter-clockwise.
6. All switches are always in the normal position with exception of the switch pairs located in the Monroeville-Greensburg region. These switches must support both normal and reverse positioning to provide vehicle "turnaround" capabilities and

allow every third vehicle passage to Greensburg and vehicle "turnaround" capabilities.

7. Vehicle speed is a constant.
   - MAGPort^R Station speed: 16 kmph
   - Switch propulsion segments: 160 kmph when switch is set divergent
   - Other propulsion segments: 400 kmph
8. A maximum of seven vehicles will be on the guideway at one time
9. Station dwell time: 2 minutes
10. Vehicle headway is a variable

We also assume that the initial starting position for the seven (7) vehicles were at the various MAGPorts with the exception of the Airport MAGPort, which does not contain a west bound vehicle. The route connection for this environment is shown in

Figure 4-1. The best objective value obtained by our genetic algorithm is 1.1389×10^5km. The convergence of the objective value is plotted in Figure 4-2. The solid curve shows the best objective values at every generation while the dotted curve shows the average objective values of the generation. From the curves, we find that the average objective value will converge to the best value as the generation increases. The string charts computed by the genetic algorithm are captured in Figure . In this illustration, the string chart demonstrates the best deadlock-free routing in this hazard-free environment.

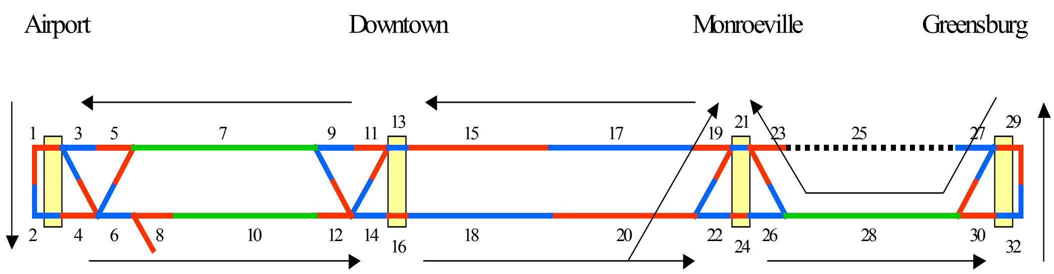

Figure 4-1: Routing in hazard-free environment.

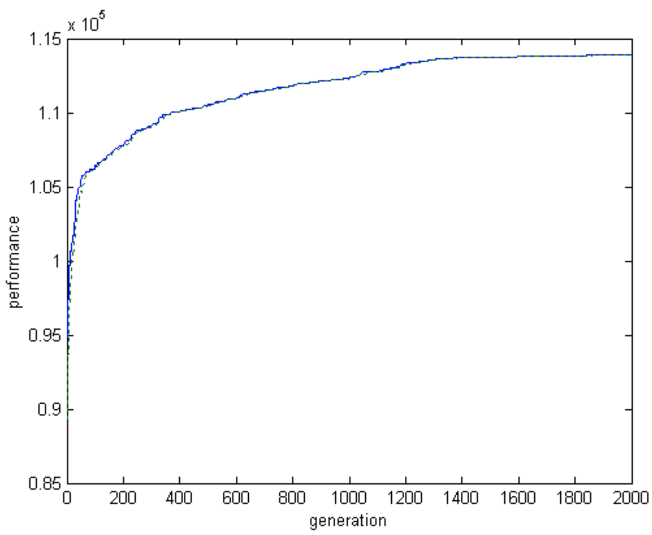

Figure 4-2: Performance objective values in hazard-free environment

4.2 Use Case II

The purpose of Use Case II is to demonstrate the routing schedule results for the deadlock and mishap-free system operations when covered hazards are present in propulsion segments that allow for these segments to be removed from service prior to any vehicle movement. In the computation, the following system constraints are considered.

1. Only one vehicle shall occupy any given propulsion segment at a given time.
2. Minimum vehicle headway shall be five minutes.
3. Minimum MAGPort dwell time shall be 30 seconds.
4. All traffic flow is counter-clockwise.
5. Vehicle speed is a constant.
   - MAGPort speed: 16 kmph
   - Switch propulsion segments: 160 kmph
   - Other propulsion segments: 400 kmph
6. Station dwell time: 2 minutes
7. Vehicle headway is a free variable.

Additional assumptions that are scenario specific will be imposed as needed.

4.2.1 Scenario 1 – Switch Hazards in Airport/Downtown Region

It is possible that the switches along the guideway may fail in a manner that prevents certain switch alignments and/or routes to be established. For example, consider the switch pair in the Airport/Downtown region of the guideway. If these switches cannot be set in the normal position, then the routing to the MAGPort^R Station "1" must be modified to accommodate this condition.

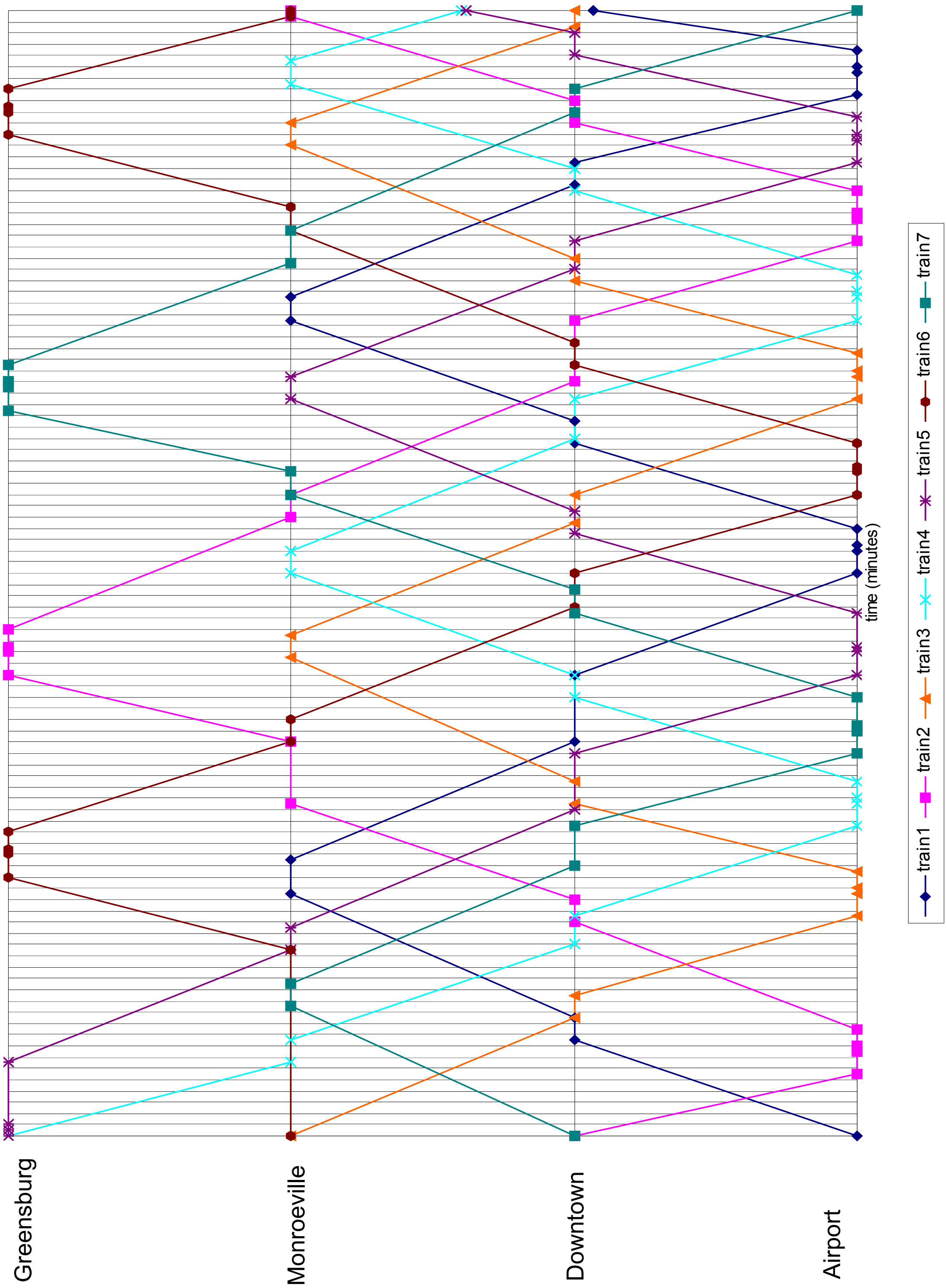

Figure 4-3: String chart in hazard-free environment.

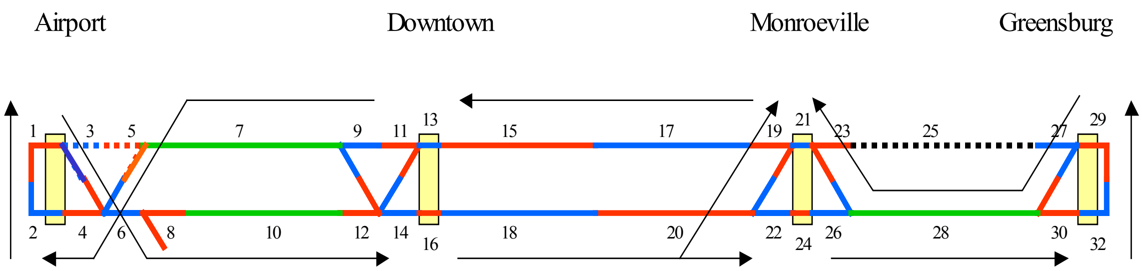

Figure 4-4: Route of switch hazards in Airport/Downtown region

In order for a vehicle to reach MAGPort^R Airport Station, the switch pair consisting of propulsion segments 5 and 6 must be set reverse and the switch consisting of propulsion segment 4 must be set normal. After the vehicle has arrived in the MAGPort^R Station 1, the vehicle must be routed to the guideway containing the even number propulsion segments to preserve the counter-clockwise vehicle movement.

Assuming that the vehicle is now departing from the Airport MAGPort^R Station, the vehicle shall have to take the divergent route. In this situation, the switch pair containing propulsion segments 3 and 4 must be set reverse and the switch pair containing propulsion segments 6 and 8 must be set normal. This configuration and the whole route connection is shown in

Figure 4-. The best objective value by our optimization algorithm is 1.1724×10^5. The convergence of the objective values is shown in Figure 4-. The string charts for the vehicles are presented in Figure 4-. For this string chart, the initial starting position for all seven (7) vehicles are at various MAGPort^R Stations with the exception of the Airport MAGPort^R Station, which contains no west bound vehicle.

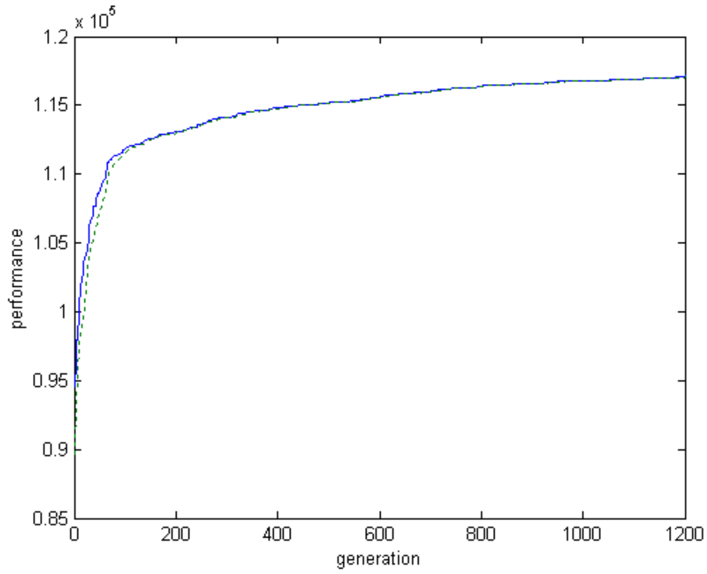

Figure 4-5: Objective Values of Switch Hazards in Airport/Downtown Region

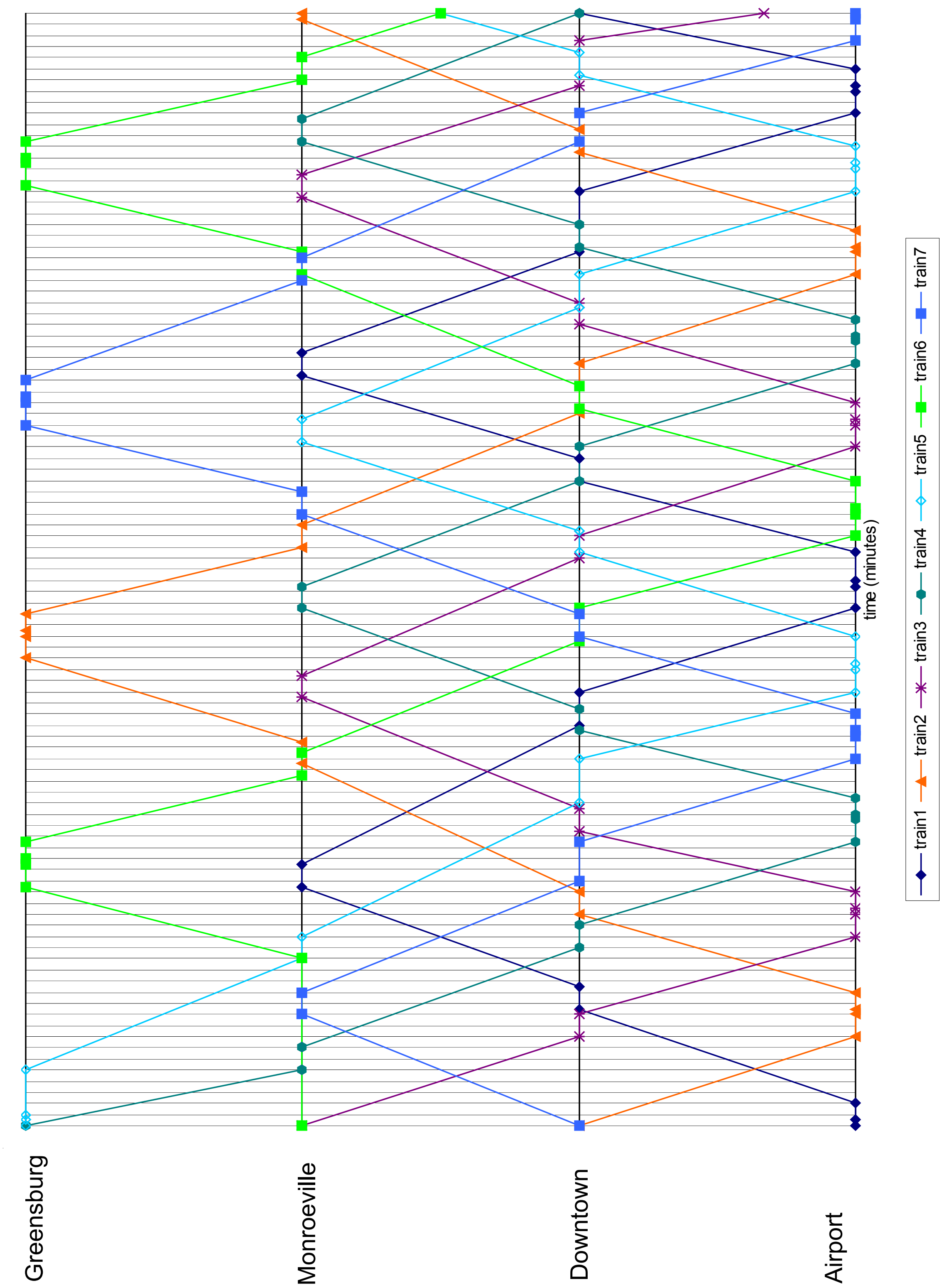

Figure 4-6: String chart of switch hazards in Airport/Downtown region

4.2.2 Scenario 2 – Airport Station Segment 1 is Out-of-Service

The MAGPort^R Station may also be Out-of-Service. For example, if the station segment 1 is in out-of-service, then the vehicles should avoid going to this segment. In order for a vehicle to reach MAGPort^R Airport Station, the switch pair consisting of propulsion segments 5 and 6 must be set reverse and the switch consisting of propulsion segment 4 must be set normal. After the vehicle has arrived at the MAGPort^R Station 2, the vehicle has to take the reverse route to departure the station. In this situation, the switch pair containing propulsion segments 4, 6 and 8 must be set normal. This configuration is shown in Figure 4-. The string charts for the vehicles are presented in Figure 4-. For this string chart, the initial starting position for all seven (7) vehicles are at various MAGPort^R Station with the exception of the Airport MAGPort^R Station, which contains no west bound vehicle.

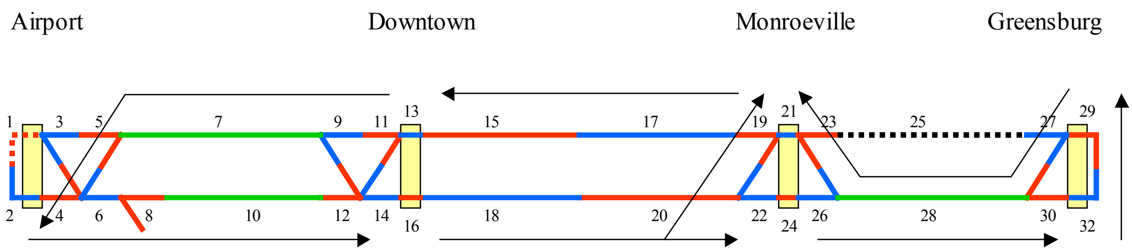

Figure 4-7: Route of segment 1 out-of-service

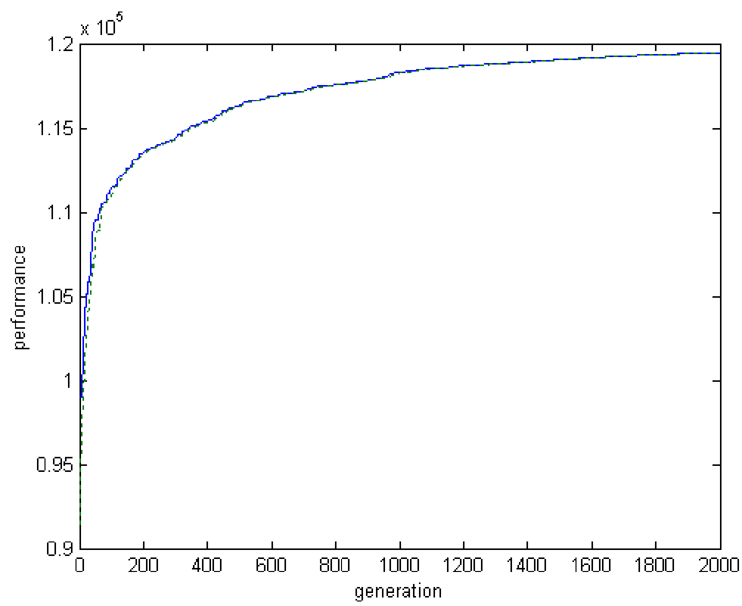

Figure 4-8: Performance objective values of segment 1 out-of-service

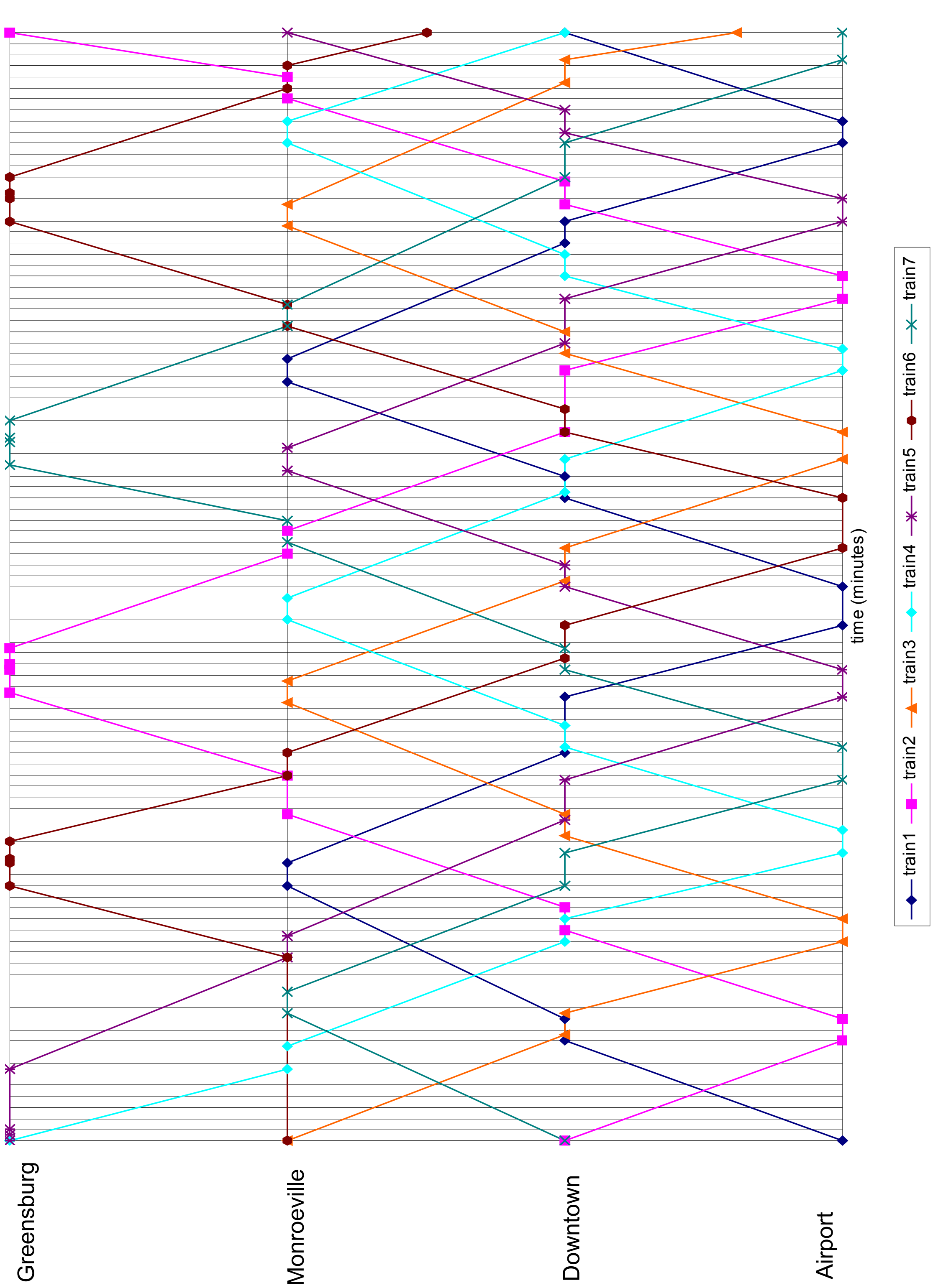

Figure 4-9: String Chart of Segment 1 Out-of-Service

4.2.3 Scenario 3 – Airport/Downtown Region Hazards

It is conceivable that propulsion segments may go out-of-service within the Airport/Downtown region of the guideway as identified in Figure 4-. If a vehicle is at the

Downtown MAGPort^R Station desiring to go to the Airport MAGPort^R Station and propulsion segment 5, 7 or 9 fails, then the vehicle must be re-routed to the parallel guideway. An example of this alternate routing is depicted in Figure 4-. In this configuration, opposing vehicles have access to the same guideway during travel. The best performance objective value obtained by our genetic algorithm is 7.2636×10^4 and the effect on system performance is captured in the string chart shown in Figure . It should be noted that in the simulation captured in this string chart only six vehicles were included, and for the included string chart, the starting positions included every MAGPort^R Station.

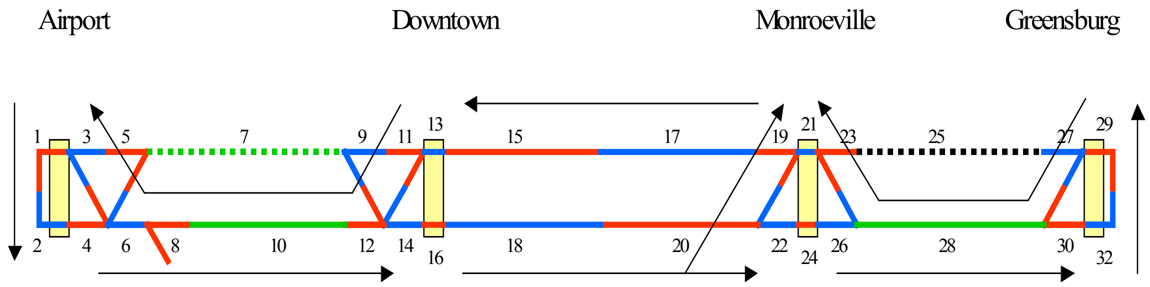

Figure 4-10: Route of segment 7 out-of-service

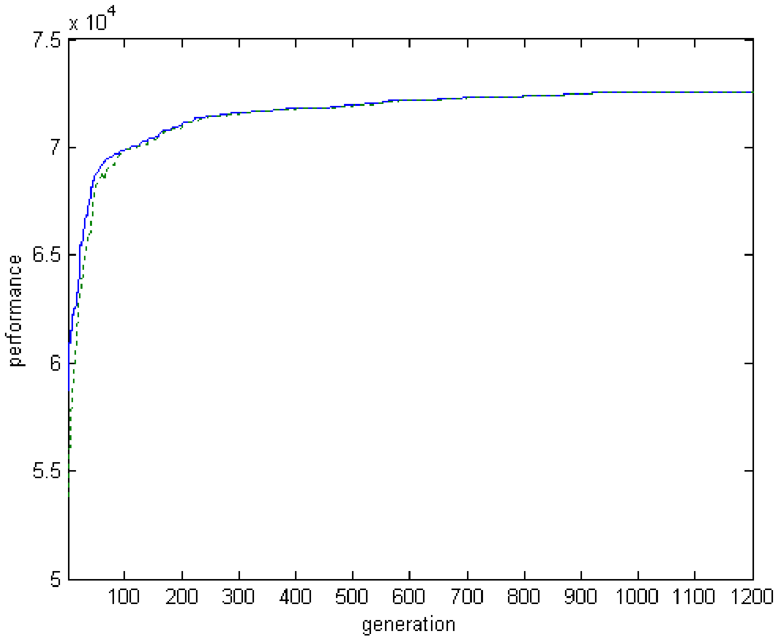

Figure 4-11: Performance objective values of segment 7 out-of-service

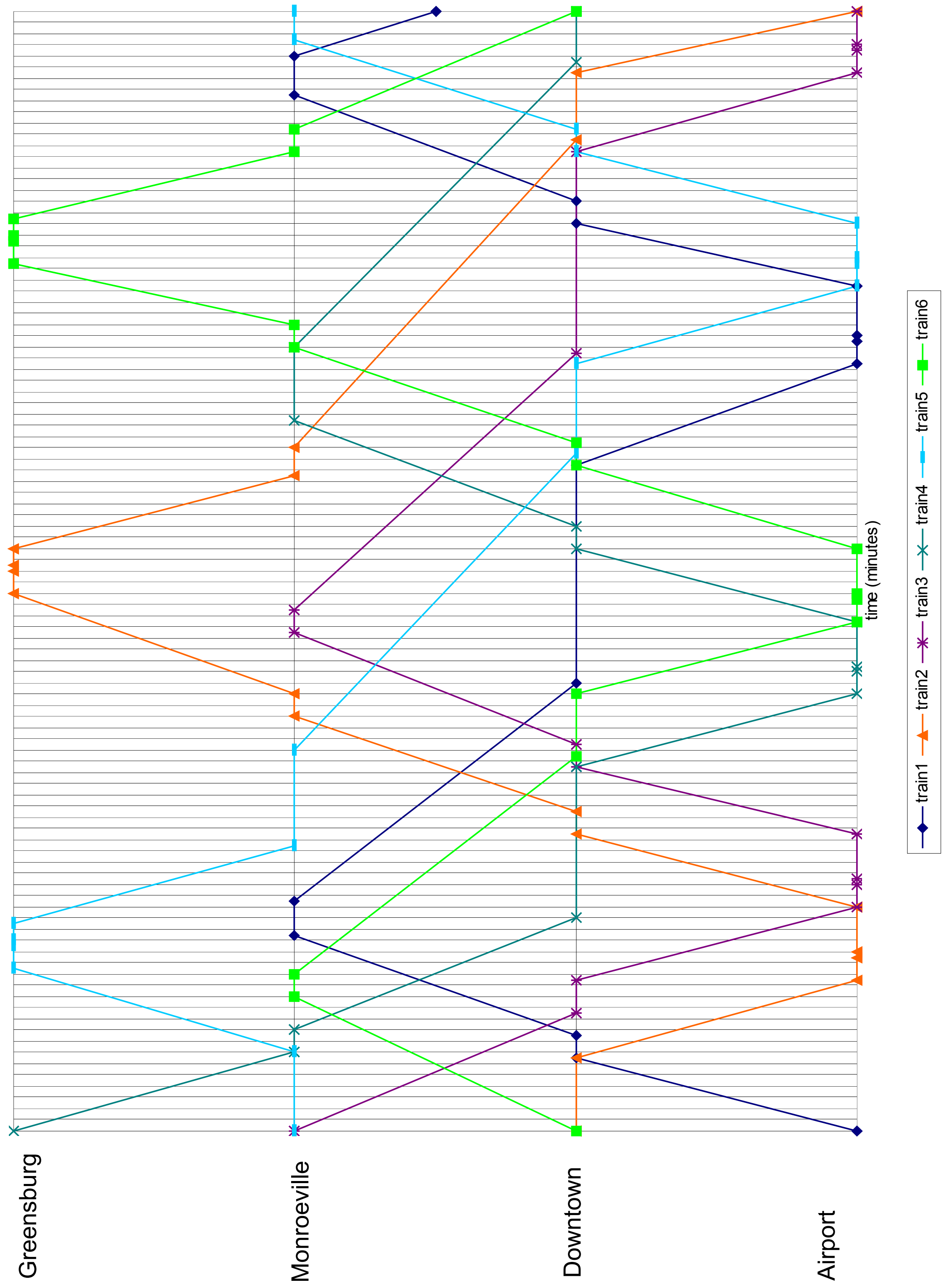

Figure 4-12: String chart of segment 7 out of service

5. Summary and Conclusions

This paper developed a vehicle routing scheduler for the Transrapid Maglev System currently under deployment in Pittsburgh, Pennsylvania. Such a scheduler maximizes the train miles traveled of the system while respecting all the operational and safety requirements. This vehicle routing scheduler takes as input the operational status of the guideway and the initial positions of the vehicles in the system, and generates a feasible routing schedule that maximizes the train miles traveled. The solution of the underlying optimization problem is based on a genetic algorithm. Several hazard-free and hazardous guideway environments were used to demonstrate the application and the effectiveness of the proposed vehicle routing scheduler. Even though the development of this scheduler is carried out in the context of the Pennsylvania Transrapid Maglev System, it is applicable to any Transrapid system to be deployed elsewhere.

References

[1] T.C. Giras, L.M. Kaufman, K.V. Ngo, D.R. Disk and F.E. Young. *Maglev Vehicle Management Algorithm (VMA) Requirements.* Final Report of the Magnetic Levitation Transportation Technology Deployment Program, submitted to Federal Railroad Administration, 2002.

[2] Holland, J.H., *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI: University of Michigan Press, 1975.

[3] Goldberg, D.E., *Genetic Algorithms in Search, Optimization and Machine Learning.* Reading, MA: Addison Wesley, 1989.

[4] Chipperfield, A.J., P.J. Fleming and H. Pohlheim, "A genetic algorithm toolbox for MATLAB". *Proc. Int. Conf. On Systems Engineering,* Coventry, UK, 6-8, 1994.

[5] Michalewics, Z. Genetic *Algorithms + Data Structures = Evolution Program.* Berlin: Springer; 1996

[6] Herrera, F., M. Lozano, and J.L. Verdegay. "Tackling real-coded genetic algorithms: oprator and tools for behavioural analysis." *Artificial Intelligence Rev.,* 12:265-319, 1998.

[7] Thangiah, S.R. and P. Petrovic. "Introduction to genetic heuristics and vehicle routing problems with complex constraints." In *Advances in Computational and Stochastic Optimization, Logic Programming, and Heuristic Search* (D.L. Woodruff, Ed.), Kluwer, New York, 253-286, 1998

[8] Jung, S. and A. Haghani. "Genetic algorithm for the time-dependent vehical routing problem." *Transportation Research Record,* 1771: 164-171, 2001

[9] Wright, A.H. "Genetic algorithms for real parameter optimization. In Foundations of Genetic Algorithms." (J.E. Rawlins,Ed.), Morgan Kaufmann, 205-218, 1991